arXiv:1203.2430v1 [q-bio.PE] 12 Mar 2012

Reconstruction of Network Evolutionary History from Extant Network Topology and Duplication History

Si LI[1], Kwok Pui CHOI[1,2], Taoyang WU[1], Louxin ZHANG[1]

[1] Department of Mathematics,
[2] Department of Statistics and Applied Probability,
National University of Singapore, Singapore 119076
{g0800874,stackp,matwt,matzlx}@nus.edu.sg

Abstract. Genome-wide protein-protein interaction (PPI) data are readily available thanks to recent breakthroughs in biotechnology. However, PPI networks of extant organisms are only snapshots of the network evolution. How to infer the whole evolution history becomes a challenging problem in computational biology. In this paper, we present a likelihood-based approach to inferring network evolution history from the topology of PPI networks and the duplication relationship among the paralogs. Simulations show that our approach outperforms the existing ones in terms of the accuracy of reconstruction. Moreover, the growth parameters of several real PPI networks estimated by our method are more consistent with the ones predicted in literature.

1 Introduction

Recent progress in experimental systems biology provides us with an unprecedented amount of genome-wide protein-protein interaction (PPI) data [9]. In order to obtain a deeper insight into the molecular machinery behind these interactions, many network models have been proposed to study or model PPI evolution [2, 17, 20]. However, PPI networks of extant organisms are only snapshots of network evolution, and inferring the whole network evolution history remains a challenging problem in computational biology [12].

Unlike many networks studied in technology and sociology, the main growth mechanism of PPI network is gene duplication and divergence [19]: when a new node is added to the network, it copies all the interactions of an existing node designed as the anchor node; subsequently some edges adjacent to one of these two nodes are randomly lost. This mechanism was explicitly converted to a network growth model by Vazquez et al. in [18]. Since then many extensions have been put forth, see for examples, [3–5, 13, 16]. Here we shall focus on a particular one called duplication-mutation with complementarity (DMC), which is the best model to fit the *D. melanogaster* (fruit fly) PPI network according to a recent study by Middendorf et al. [11].

When a growth model is fixed, the problem of reconstructing the evolutionary history of an observed network is to infer the relative order of the nodes according to which the network evolved (see Section 2.2 for definitions). Better understanding of this problem can provide further insights into not only how PPI networks are formed, but also how they will possibly evolve in the future. Several approaches to address this problem have been proposed in recent years. In order to obtain better ways of predicting protein modules, Dutkowski and Tiuryn introduced a Bayesian network framework to infer the posterior probability of interactions between ancestral nodes based on a duplication and speciation model [6]. A similar approach was used by Pinney [15] to infer ancestral interactions between bZIP proteins. Gibson and Goldberg proposed a merging algorithm to reconstruct the evolutionary history of PPI networks using gene trees [8]. A novel framework for estimating the topology of the ancestral networks based on maximal likelihood was presented by Navlakha and Kingsford in [12]. Recently, Patro et al. [14] used a maximal parsimony approach that appends edges in observed networks to duplication history forest.

Here we introduce a new history inferring framework based on the maximal likelihood principle. In contrast to the model-based methods in [12], our approach incorporates not only the topology of observed networks, but also the duplication history of the proteins contained in the networks. Although the evolution of topology is often determined by some growth mechanisms, the duplication history of the proteins can be inferred independently by phylogenetic studies [14, 15]. After establishing some theoretical results concerning the DMC model, we reduce the problem of finding most probable history of ancient networks to an optimization problem, and propose some efficient heuristic algorithms to solve the latter problem. Simulations show that our method provides better inference than the ones in [12]. Moreover, we also applied our algorithm to the PPI networks of *S. cerevisiae* (budding yeast), *D. melanogaster* and *C. elegans* (worm), and the growth parameters obtained by our approach are more consistent with the ones predicted in [7, 19]. Finally, we also propose an improved measure for comparing two histories.

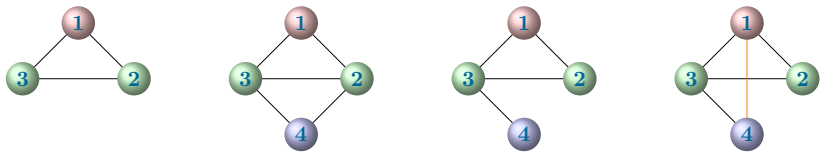
The rest of the paper is organized as follows: Section 2 provides the framework of reconstruction, including the technical background and the inference method. In Section 3 we present the inference results for simulations and real data sets. We conclude in Section 4 with a brief discussion and some possible related research directions.

2 Methods

2.1 Modeling Network Evolution

In the DMC model $\mathcal{M} := \mathcal{M}(p_c, p)$, where p_c and p are the two parameters that specify the model, we start with an initial graph G_0, the so-called seed graph. At each time step t, the graph G_t is obtained from G_{t-1} by the following procedures (see Fig. 1 for an illustration): (1) (Duplication) A node u_t is chosen uniformly at random from the set of nodes in G_{t-1}, and a new node v_t is added and connected

to every neighbor of u_t. Here u_t and v_t are often referred to as the anchor node and duplicate node at step t, respectively. (2) (Mutation) For each neighbor of u_t, say w, we choose one edge from (u_t, w) and (v_t, w) with equal probability, and this chosen edge is deleted with probability $1 - p$. (3) (Complementarity) The nodes u_t and v_t are connected with probability p_c.



(A) (B) (C) (D)

Fig. 1: Illustration of the DMC model. (B) is obtained from (A) by one duplication step, with node 1 (represented in maroon) as the anchor node and node 4 as the duplicate node (represented in purple); the probability that node 1 is chosen as the anchor node is 1/3 because the network in (A) contains three nodes. (C) is obtained from (B) by the mutation step, which occurs with probability $p(1-p)/2$. (D) is obtained from (C) by the complementarity step,which occurs with probability p_c.

Note that the DMC model is Markovian, that is, the probability of obtaining G_t when G_{t-1} is given depends solely on the parameters of $\mathcal{M}$. For example, denoting the network (A) and (D) in Fig 1 by G_{t-1} and G_t, respectively, then the probability $\mathbb{P}(G_t|G_{t-1}, \mathcal{M})$ that G_t is evolved from G_{t-1} by one step under the model $\mathcal{M}$ is $p(1-p)p_c/2$.

2.2 History Reconstruction

Given an observed network G, a *growth history* $\mathcal{H}$ of G is a graph sequence $(G_0, G_1, \cdots, G_n)$ such that $G_n = G$ and for each index t in $\{1, \cdots, n\}$, the graph G_t can be obtained from G_{t-1} in one step under the DMC model $\mathcal{M}$. The first graph G_0 is referred to as the seed graph of the history. In addition, the number n is called the *span* of the history. Clearly, a history $\mathcal{H}$ induces a unique sequence $\theta := \theta(\mathcal{H})$ of duplicate nodes, that is, $\theta = (v_1, \cdots, v_n)$ such that for all t, node v_t is the unique node in G_t, but not G_{t-1}.

Given a network G, let $\mathcal{H}$ be the growth history we hope to infer. The probability of G being evolved according to history $\mathcal{H}$, when viewed as a function of the unknown history $\mathcal{H}$, is called the *likelihood function* $L(\mathcal{H} \,|\, G, \mathcal{M})$ that is given by

$$L(\mathcal{H} \,|\, G, \mathcal{M}) = \prod_{t=1}^{n} \mathbb{P}(G_t|G_{t-1}, \mathcal{M}).$$

We adopt a maximal likelihood approach to infer the history of G as below.

Problem 1. Given a network G together with a natural number n and model $\mathcal{M}$, construct a growth history $\mathcal{H}$ that maximizes the likelihood $L(\mathcal{H} \mid G, \mathcal{M})$ among all histories with span n.

This problem is expected to be difficult since the number of possible histories grows exponentially, and we are not aware of any results concerning whether this problem is polynomial-time solvable. Before introducing a variant of the above problem that is more tractable, we present some necessary tools in the following two subsections.

2.3 Duplication Forest

We begin with duplication history, which is closely related to network history as gene duplication is a major driving force of PPI network evolution [19]. The idea of encoding the duplication history by a forest of binary tree was used in [12, 14]. Patro et al. [14] incorporated duplication history in a parsimony approach to reconstruct network history.

A growth history $\mathcal{H}$ of a PPI network induces a unique duplication forest. Initially, we have a forest Γ_0 consisting of isolated nodes that are identical to the set of nodes in the seed graph. At each step t, the forest Γ_t is obtained from Γ_{t-1} by replacing the anchor node u_t with a cherry $\{u_t, v_t\}$ consisting of u_t and the duplicate node v_t. Here a *cherry* $\{u, v\}$ is referred to a subtree consisting of two leaves u and v and the internal node adjacent to them.

The duplication forest of a PPI network can also be inferred independently without using its growth history. For instance, such a forest can be reconstructed by the phylogenetic relationships between the genes in the network [15]. This observation is key to our investigation.

2.4 Backward Operator

In this subsection, we will introduce a backward operator that is important in our inference framework.

Consider one step in a growth history, that is, a graph G_t obtained from G_{t-1} in one step by using anchor node u_t and duplicate node v_t. Now we want to define a backward operator $\mathcal{R}$ such that G_{t-1} can be determined by this operator and the triplet (G_t, u_t, v_t). To this end, let $\mathcal{R}_{v_t}^{u_t}(G_t)$ be the graph obtained from G_t by merging the two nodes u_t and v_t in G_t, that is, (i) for each neighbor w of v_t such that $w \neq u_t$ and w is not adjacent to u_t, add an edge (w, u_t); (ii) delete the node v_t and all edges incident to it.

Similarly, the backward operator can be applied to the duplication forest, that is, $\mathcal{R}_{v_t}^{u_t}(\Gamma_t)$ is the forest obtained from Γ_t by replacing the cherry $\{u_t, v_t\}$ with the leaf u_t. Note that this definition is consistent with the above one in the following sense: If Γ_t is the duplication forest corresponding to the network G_t, then $\mathcal{R}_{v_t}^{u_t}(\Gamma_t)$ is the duplication forest associated with $\mathcal{R}_{v_t}^{u_t}(G_t)$. When the anchor node u_t is clear from the context, we also write $\mathcal{R}_{v_t}$ for $\mathcal{R}_{v_t}^{u_t}$.

2.5 Growth History with Known Duplication Forest

Using the backward operator introduced above, we shall introduce a scheme to represent a growth history with known duplication forest by a node sequence. Throughout this paper, we use the convention that a node sequence consists of distinct nodes, while a node list may contain repeated nodes.

In general, a node sequence $\theta = (v_1, \cdots, v_n)$ and a duplication forest Γ are said to be *compatible* if there exists a (necessarily unique) sequence $(\Gamma_1^\theta, \cdots, \Gamma_n^\theta)$ of forests such that $\Gamma_n^\theta = \Gamma$, and $\Gamma_{t-1}^\theta = \mathcal{R}_{v_t}(\Gamma_t)$ holds for each $t \in \{1, \cdots, n\}$. Note that a necessary and sufficient condition for θ and Γ being compatible is that v_t belongs to a cherry in Γ_t^θ for each t. Denoting the sibling of v_t in Γ_t^θ, that is, the unique leaf in Γ_t that forms a cherry with v_t, by u_t, we say the list $\pi = (u_1, \cdots, u_n)$ is the *anchor* list determined by Γ and θ.

As mentioned above, a growth history $\mathcal{H} = (G_0, \cdots, G_n)$ specifies a duplicate sequence $\theta = (v_1, \cdots, v_n)$ and a duplication forest Γ. Clearly, the sequence θ and forest Γ must be compatible. On the other hand, given a duplication forest Γ associated with a network G and a sequence θ that is compatible with Γ, then there exists a unique growth history $\mathcal{H}$ such that θ is induced from $\mathcal{H}$. In other words, when the duplication forest Γ is fixed, a growth history $\mathcal{H}$ is uniquely determined by the duplicate sequence θ associated with it. In this context, the likelihood function is defined as

$$L(\theta \,|\, G, \Gamma, \mathcal{M}) := \prod_{i=1}^{n} \mathbb{P}(G_i^\theta \,|\, G_{i-1}^\theta, \Gamma, \mathcal{M}),$$

where $\mathbb{P}(G_i^\theta | G_{i-1}^\theta, \Gamma, \mathcal{M})$ is the probability that G_i^θ is evolved from G_{i-1}^θ in one step under the DMC model $\mathcal{M}$ and using the anchor node u_t specified by θ and Γ. Note that in general the probability $\mathbb{P}(G_i^\theta | G_{i-1}^\theta, \Gamma, \mathcal{M})$ is different from $\mathbb{P}(G_i^\theta | G_{i-1}^\theta, \mathcal{M})$. Indeed, the latter can be regarded as an "average" of the former over all possible anchor nodes.

Now, the problem of inferring growth history with given duplication forest, a variant of Problem 1 that will be studied in this paper, can be formally stated as below.

Problem 2. Given a network G together with a duplication forest Γ and a growth model $\mathcal{M}$, construct a duplicate sequence θ such that the likelihood $L(\theta \,|\, G, \Gamma, \mathcal{M})$ is maximized.

In the above problem, the parameters in the DMC model $\mathcal{M}$ are specifically mentioned. However, as we shall see later, the parameters of $\mathcal{M}$ are not needed for the history inference problem.

2.6 Theoretical Results

Here we present some theoretical results that are crucial to solve Problem 2. Due to space limitations, all proofs are outlined in the Appendix.

Lemma 1. *Given a network G with duplication forest Γ, for any two sequences θ_1 and θ_2 that are compatible with Γ, the graph $G_0^{\theta_1}$ is isomorphic to $G_0^{\theta_2}$.*

Given a duplicate sequence $\theta = (v_1, v_2, \cdots, v_n)$, we shall associate it with three families of numbers that are crucial to our analysis. For each duplicate node v_i in θ, let $\delta(v_i)$ be the indicator function that takes value 1 if v_i is connected to its anchor node u_i, and 0 otherwise; $\alpha(v_i)$ the number of the neighbors shared by v_i and u_i; and $\beta(v_i) := \beta(v_i, G_i^\theta)$ the number of nodes adjacent to v_i or u_i in G_i^θ, but not both. Note that $2\delta(v_i) + 2\alpha(v_i) + \beta(v_i)$ is equal to the sum of the degree of v_i and that of u_i in G_i^θ.

The sum $\delta(\theta) := \sum_{i=1}^n \delta(v_i)$ is called the *complementarity number* of history θ, the sum $\alpha(\theta) := \sum_{i=1}^n \alpha(v_i)$ is called the *extension number* of θ, and $\beta(\theta) := \sum_{i=1}^n \beta(v_i)$ is called the *loss number* of θ.

We complete this subsection with the following two key results. The first one shows that the complementarity number and extension number are constants over all compatible duplicate sequences.

Theorem 1. *Given a network G with duplication forest Γ and two compatible duplicate sequences θ_1 and θ_2, we have $\delta(\theta_1) = \delta(\theta_2)$ and $\alpha(\theta_1) = \alpha(\theta_2)$.*

Theorem 2. *Given a network G with duplication history Γ, the ratio of two likelihood functions for two duplicate sequences θ_1 and θ_2 that are compatible with Γ is given by*

$$\frac{L(\theta_1 \,|\, G, \mathcal{M}, \Gamma)}{L(\theta_2 \,|\, G, \mathcal{M}, \Gamma)} = \Big(\frac{1-p}{2}\Big)^{\beta(\theta_1)-\beta(\theta_2)}.$$

2.7 Reconstruction Algorithms

By Theorem 2, solving Problem 2 is equivalent to solving the following problem.

Problem 3. Given a network G and its duplication forest Γ, construct a duplicate sequence θ such that the loss number $\beta(\theta)$ is minimized among all sequences compatible with Γ.

In this section, we propose some heuristic algorithms to solve Problem 3, and hence Problem 2. The first one is a greedy algorithm called minimal loss number (MLN), in which we choose a duplicate node with the smallest value $\beta(v)$ among all candidate ones.

To motivate our main reconstruction algorithm, we introduce some further notation and results. A duplicate sequence $\theta_1 = (v_1, \cdots, v_n)$ is said to be swapped from $\theta_2 = (v_1', \cdots, v_n')$ at position m for some index $m \in \{1, \cdots, n-1\}$ if we have $v_m' = v_{m+1}$, $v_{m+1}' = v_m$, and $v_i' = v_i$ for all other indices i.

Lemma 2. *Given a network G with duplication forest Γ, if θ_1 and θ_2 are two compatible duplicate sequences such that θ_1 is swapped from θ_2 at position m, then we have $G_i^{\theta_1} = G_i^{\theta_2}$ for each index $i \in \{0, \cdots, n\}$ with $i \neq m$.*

Let θ_1 and θ_2 be two compatible duplicate sequences as stated in the above lemma. By Lemma 2 and Theorem 2, $L(\theta_1 \,|\, G, \Gamma, \mathcal{M}) \geq L(\theta_2 \,|\, G, \Gamma, \mathcal{M})$ if and only if for $G_m = G_m^{\theta_1} = G_m^{\theta_2}$, we have

$$\beta(v_m, G_m) + \beta(v_{m-1}, \mathcal{R}_{v_m}(G_m)) \leq \beta(v_{m-1}, G_m) + \beta(v_m, \mathcal{R}_{v_{m-1}}(G_m)). \quad (1)$$

Motivated by the above observation, for two cherries $\{u, v\}$ and $\{u', v'\}$ in Γ_t, we say $\{u, v\}$ is more *favorable* than $\{u', v'\}$, denoted by $\{u, v\} \succ \{u', v'\}$, if $\beta(v, G_t) + \beta(v', \mathcal{R}_v^u(G_t)) < \beta(v', G_t) + \beta(v, \mathcal{R}_{v'}^{u'}(G_t))$ holds. Note that in general the relation $\succ$ is not transitive, that is, $\{u, v\} \succ \{u', v'\}$ and $\{u', v'\} \succ \{u^*, v^*\}$ does not imply $\{u, v\} \succ \{u^*, v^*\}$.

Now we present our main inference algorithm called cherry greedy (CG), which runs as follows: At every backward reconstruction step, we choose a node from the most favorable cherry C, that is, the number of cherries C' with $C \succ C'$ is maximized. If several cherries are equally favorable, we uniformly choose one of them. More precisely, starting from $G_t := G$ and $\Gamma_t := \Gamma$, we choose a most favorable cherry (u, v) from Γ_t and uniformly choose one node from the cherry, say v_t, as the duplicate node at this step. Then we construct G_{t-1} as $\mathcal{R}_{v_t}(G_t)$ and $\Gamma_{t-1} = \mathcal{R}_{v_t}(\Gamma_t)$. This process continues until G_0 is obtained.

Since the above algorithm is a stochastic one, that is, among a chosen cherry $\{u, v\}$, u and v has the equal probability of being chosen as the duplicate node. Therefore, one natural way of improving its accuracy is to repeat the algorithm for a certain times and report the best output, where the number of repetitions can be regarded as a tuning parameter. When the real duplicate sequence θ_{real} is known, the best one is defined as the output θ such that Kendall's τ between θ_{real} and θ is maximized (see Section 3 for further details on Kendall's τ), otherwise the one with the smallest loss number is chosen. This strengthened version of the CG algorithm with be refereed to as CGR, where 'R' stands for repetition.

2.8 Estimating Parameters

From the results in Section 2.6 and Section 2.7, it is clear that the parameters of the DMC model are not used in our inference framework. Moreover, here we will present a method by which the parameters can be established after a growth history being inferred.

To this end, assume that a growth history $\mathcal{H} = (G_0, \cdots, G_n)$, together with the duplicate sequence $(v_1, \cdots, v_n)$ and anchor list $(u_1, \cdots, u_n)$, is given. Note that for each neighbor w of node u_i in G_{i-1}, the probability that w is adjacent to both u_i and v_i in G_i is p. In other words, the extension number $\alpha(v_i)$ at i-th step, i.e., the number of the common neighbors shared by u_i and v_i in G_i, has the binomial distribution with parameters p and $\beta(u_i) + \alpha(v_i)$, where $\beta(u_i) + \alpha(v_i)$ is the number of neighbors that u_i has in G_{i-1}. On the other hand, the random variable $\delta(v_i)$ has Bernoulli distribution with parameter p_c. Therefore, we are led to propose the estimators $\hat{p} = \frac{\alpha(\theta)}{\beta(\theta)+\alpha(\theta)}$ and $\hat{p}_c = \frac{\delta(\theta)}{n}$ to estimate the parameters p and p_c respectively.

3 Results

Our reconstructing algorithms, minimal loss number (MLN) and cherry greedy (CG), have been implemented in Perl, which is available upon request. Given a network G and duplication forest Γ, each outputs a hypothetical duplicate sequence θ that approximates the one with the minimal loss number.

To assess the performance, we need to measure the difference between the inferred duplicate sequence and the 'real' one. One popular index for this purpose is Kendall's tau K_τ [1, 12]. Formally, for two sequences $\theta_1 = \{v_1, \cdots, v_n\}$ and $\theta_2 = \{v'_1, \cdots, v'_n\}$ that consist of the same set of nodes, $K_\tau(\theta_1, \theta_2)$ is defined as

$$K_\tau(\theta_1, \theta_2) = \frac{2(n_c - n_d)}{n(n-1)},$$

where n_c is the number of concordant pairs, that is, the number of pairs in θ_1 that are in the correct relative order with respect to θ_2,and n_d is the number of discordant pairs. Note that we have $K_\tau(\theta_1, \theta_2) = 1$ if the two sequences are identical, and $K_\tau(\theta_1, \theta_2) = -1$ if they are exactly opposite.

3.1 Simulation Validation

To validate our algorithms, we generated 100 random network using each DMC model $\mathcal{M}$, where the parameters p_c and p ranged from 0.1 to 0.9 at 0.2 intervals. Each network has 100 nodes and is evolved from the same seed graph K_2 (i.e., the graph with two nodes and one edge).

For each simulated network G, its duplication forest Γ and duplicate sequence θ_{real} were recorded. Next, we reconstructed duplicate sequences using our algorithms. The one using MLN is denoted by θ_{MLN}, and the one using CG by θ_{CG}. We also considered the algorithm CGR, which outputs θ_{CGR}, the one with the highest Kendall's τ among ten runs of CG. We ran some of the experiments more than 10 times but found that more runs did not improve the results much, and hence we ran 10 times throughout. For comparison, we also generated a random duplicate sequence θ_{rand}, which can be interpreted as a 'null model'. Finally, we computed $K_\tau(\theta_{\text{real}}, \theta)$ for $\theta \in \{\theta_{\text{rand}}, \theta_{\text{MLN}}, \theta_{\text{CG}}, \theta_{\text{CGR}}\}$.

The results for $K_\tau(\theta_{\text{real}}, \theta_{\text{rand}})$ and $K_\tau(\theta_{\text{real}}, \theta_{\text{MLN}})$ are summarized in Fig. 2. Our results for $K_\tau(\theta_{\text{real}}, \theta_{\text{rand}})$ agree well with the theoretical mean of $K_\tau(\theta_{\text{real}}, \theta_{\text{rand}})$, which is 0. In addition, the results for $K_\tau(\theta_{\text{real}}, \theta_{\text{CG}})$ and $K_\tau(\theta_{\text{real}}, \theta_{\text{CGR}})$ are summarized in Fig. 3. From these results, we can see that compared to random duplicate sequences, our algorithms have improved the values of Kendall's τ substantially. In addition, in general CG has better performance than MLN. Finally, repeating algorithm CG a few times will increase the performance.

3.2 Comparison with Existing Methods

In this subsection, we compare the performance of our algorithm CG with NetArch, the inference method introduced in [12]. Since duplication forest is

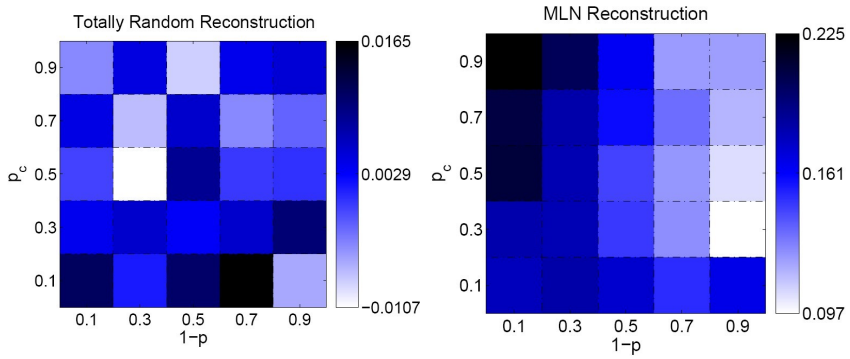


Fig. 2: Results for simulation data sets. The figure in the left is the heat map representing the values of $K_\tau(\theta_{\text{real}}, \theta_{\text{rand}})$, and the one in the right is for $K_\tau(\theta_{\text{real}}, \theta_{\text{MLN}})$. Here the value of Kendall's τ is represented by the intensity of color.

not incorporated in the framework proposed in [12], it would be expected that CG will outperform NetArch.

Indeed, Fig. 3 already shows that our algorithm CGR outperforms NetArch because in [12], the authors claimed that the values of Kendall's τ between the real duplicate sequence and the one constructed by their method are between 0.2 and 0 for the same set of combinations of parameters.

Even without using repetition, CG also outperforms NetArch in general. We demonstrate this by comparing the performance of them over 100 simulated random networks. For each simulation, we generated a pair of parameters p and p_c uniformly from the interval $(0, 1)$, and one graph G with 30 nodes from the seed graph K_2 using the DMC model $\mathcal{M}$. As above, the duplication forest Γ and duplicate sequence θ_{real} were recorded. Next, both NetArch and CG were used to reconstruct the duplicate sequence, and their outputs were denoted by θ_{Net} and θ_{CG} history. Finally, the values $\tau_1 := K_\tau(\theta_{\text{real}}, \theta_{\text{CG}})$ and $\tau_2 := K_\tau(\theta_{\text{real}}, \theta_{\text{Net}})$ were computed.

Among the 100 simulated networks, CG outperforms NetArch 87 times, and the distributions of $\tau_2 - \tau_1$ and $\tau_1 - \tau_2$ are summarized in Fig. 4a. Note that for the cases when CG outperforms NetArch, the gains in terms of Kendall's tau is significant, i.e., the average value is 0.2.

Moreover, we also compared the parameters $\hat{p}$ and $\hat{p_c}$ estimated by using CG with the ones p^{best} and p_c^{best} obtained by the method in [12]. Fig.4b are the box plots for the errors of these four estimations, in which the data are calculated as $|p - \hat{p}|$, etc. Note that the closer to 0, the better the estimation is. We can see that our method has smaller means of errors and smaller length of confidence intervals for both p and p_c.

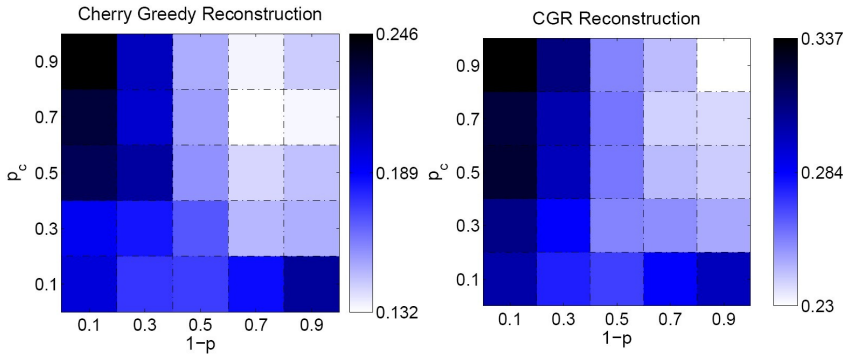


Fig. 3: Results for the algorithm CG and CGR. The figure in the left is the heat map for $K_\tau(\theta_{\text{real}}, \theta_{\text{CG}})$ and the one in the right for $K_\tau(\theta_{\text{real}}, \theta_{\text{CGR}})$. In CGR we run CG for 10 times and report the output with the highest Kendall's τ.

3.3 Application to Real PPI Networks

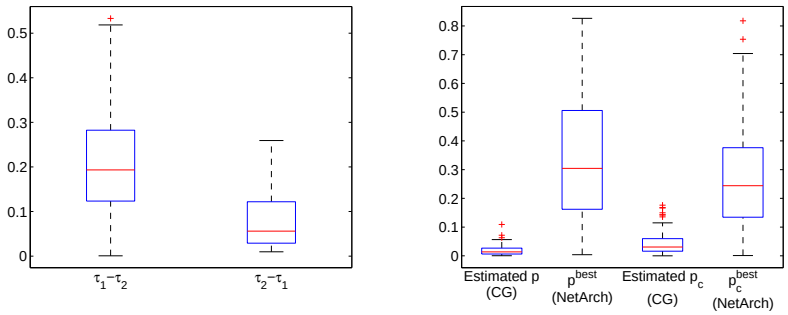
We downloaded 460 gene trees reconciled in [6]. The gene trees contain genes from *S. cerevisiae* (budding yeast), *D. melanogaster* (fruit fly) and *C. elegans* (worm). For each gene tree, we used the genes of one species and deleted all the genes from the other two species to create a gene duplication forest for each species. In addition, we downloaded corresponding PPI networks from the database DIP (http://dip.doe-mbi.ucla.edu/dip/Main.cgi). Since the gene trees obtained in this way are timed, we can infer from them a duplicate sequence θ^*_{real} that approximates the real duplicate sequence.

When we checked the gene trees, we found that some of them, especially the large ones, are very asymmetric about the root, which are not common for the duplication trees associated to networks generated by the DMC model. To handle this asymmetry, we modified our inference algorithm CG by taking account the depth of leaves (i.e., the number of edges between the leave and the root). More precisely, in each backward step we choose the most favorable cherry among the cherries whose depth is larger than a threshold. The output of this modified CG algorithm will be denoted by θ^*_{CG}.

The values of $\tau = K_\tau(\theta^*_{\text{real}}, \theta^*_{\text{CG}})$ for the three networks are listed in Table 1. In addition, the corresponding estimated parameters $\hat{p}$ and $\hat{p}_c$ are also listed. Note that these estimations are consistent with those in [7, 19], where the authors asserted that p and p_c are smaller than 0.1. Since the one obtained in [12] is 0.7, here we also demonstrate the advantage of incorporating duplication history in growth history reconstruction.

3.4 An improved measure

Typically one cannot distinguish between a duplicate node from its anchor node. Therefore, while Kendall's tau between two sequences is natural for comparing



(a) (b)

Fig. 4: (a) Box plot for differences between two methods. τ_1 is the Kendall τ obtained by CG and τ_2 is obtained by NetArch. For $\tau_1 - \tau_2$, we only consider the cases $\tau_1 > \tau_2$, and likewise for $\tau_2 - \tau_1$. (b) Box plot for errors of estimations of parameters. Here parameters are uniformly generated from the interval $(0, 1)$.

Table 1: The Kendall's τ and estimated parameters for three PPI networks.

	S.cerevisiae	*C. elegans*	*D. melanogaster*
$\hat{p}$	0.061142	0.020976	0.025953
$\hat{p}_c$	0.053215	0.048443	0.024182
τ	0.378	0.316	0.473

duplicate sequence, it also inherits the intricate difficulty of separating anchor nodes from duplicate nodes. To overcome this problem, we propose an alternative measure to compare two duplicate sequences, by which the 'symmetry' between anchor nodes and duplicate nodes is taken into account.

To begin with, each internal node of the duplication forest Γ is labeled by a unique label. Note that each duplicate sequence θ that is compatible with Γ induces a unique sequence $\gamma(\theta)$ by replacing duplicate node v_i with the label of the parent of v_i in Γ_i^θ. For two duplicate sequences θ_1 and θ_2, let $K_\tau^*(\theta_1, \theta_2) := K_\tau(\gamma(\theta_1), \gamma(\theta_2))$, and we argue this is a more appropriate measure since here we do not make a distinction between anchor nodes and duplicate nodes. Using the simulated networks obtained in Section 3.1, we present in Fig. 5 the results for $K_\tau^*(\theta_{\text{real}}, \theta_{\text{comp}})$ and $K_\tau^*(\theta_{\text{real}}, \theta_{\text{CG}})$, where θ_{comp} is a duplicate sequence uniformly chosen from all compatible sequences. These results also validate our algorithm CG as $K_\tau^*(\theta_{\text{real}}, \theta_{\text{CG}})$ is higher than $K_\tau^*(\theta_{\text{real}}, \theta_{\text{comp}})$.

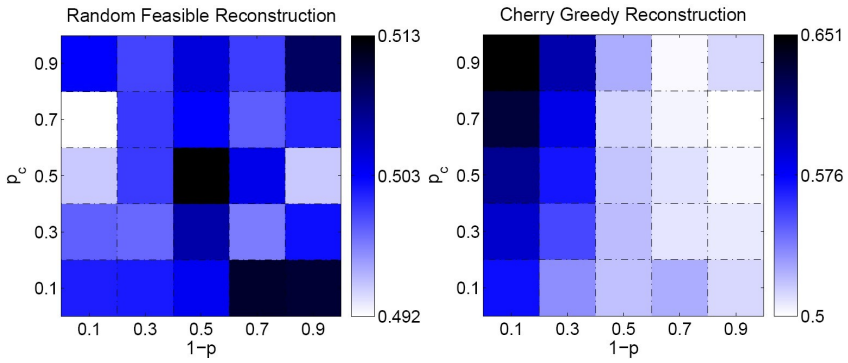


Fig. 5: Results measured by K_τ^*. The figure in the left is for $K_\tau^*(\theta_{\text{real}}, \theta_{\text{comp}})$ and the one in the right for $K_\tau^*(\theta_{\text{real}}, \theta_{\text{CG}})$. Here the simulated networks are the same as the ones used in obtaining Fig. 2.

4 Discussion

Assuming the observed network is the result of a growing mechanism as depicted in the DMC model, we have presented a likelihood-based algorithm for recovering the most probable network evolutionary history by exploiting the known duplication history trees of paralogs in the observed network. Through a series of reduction of the search space of all histories to (i) compatible duplicate sequences and (ii) the set of favored duplicate nodes, we have provided a computationally efficient algorithm. Our approach successfully re-traces the network evolution especially in the scenario that the labels of ancestor nodes are not necessarily to be one of the duplicates. As a useful by-product of our reconstruction, we propose natural estimators for the model parameters which are of independent interest. Our approach can be applied to infer the order of duplication events and to trace the topological characteristics of networks as they evolve. Our method, though described in the context of the DMC model, can be adapted to other network growing models. In addition, it can potentially be extended to predict the emergence of interactions and modules during the network evolution, and hence to provide comparison of the evolution history across different species.

Acknowledgments This work is supported from the Singapore MOE grant R-146-000-134-112. We are grateful to Dr. Navlakha and Kingsford for providing the code in [12].

References

[1] J. Bar-Ilan, M. Mat-Hassan, and M. Levene (2006) Methods for comparing rankings of search engine results. *Comput. Netw.*, 50:1448–1463.

[2] A. Barabasi and Z. Oltvai (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, 5:101–113.

[3] G. Bebek, P. Berenbrink, C. Cooper, T. Friedetzky, J. Nadeau, and S. Sahinalp (2006) The degree distribution of the generalized duplication model. *Theor. Comp. Sci.*, 369:234–249.

[4] A. Bhan, D. Galas, and T. Dewey (2002) A duplication growth model of gene expression networks. *Bioinformatics*, 18:1486–1493.

[5] F. Chung, L. Lu, T. Dewey, and D. Galas (2003) Duplication models for biological networks. *J. Comput. Biol.*, 10:677–687.

[6] J. Dutkowski and J. Tiuryn (2007) Identification of functional modules from conserved ancestral protein-protein interactions. *Bioinformatics*, 23:i149–i158.

[7] N. Farid and K. Christensen (2006) Evolving networks through deletion and duplication. *New J. Phys.*, 8:212–229.

[8] T. Gibson and D. Goldberg (2009) Reverse engineering the evolution of protein interaction networks. *Pac. Symp. Biocomp.*, pp 190–202.

[9] L. Hakes, J. Pinney, D. Robertson, and S. Lovell (2008) Protein-protein interaction networks and biology–what's the connection. *Nat. Biotech.*, 26:69–72.

[10] I. Ispolatov, P. Krapivsky, and A. Yuryev (2005) Duplication-divergence model of protein interaction network. *Phys. Rev. E*, 71:061911.

[11] M. Middendorf, E. Ziv, and C. Wiggins (2005) Inferring network mechanisms: The *drosophila melanogaster* protein interaction network. *Proc. Natl. Acad. Sci.*, 109:3192–3197.

[12] S. Navlakha and C. Kingsford (2011) Network archaeology: Uncovering ancient networks from present-day interactions. *PLoS Comput. Biol.*, 7:e1001119.

[13] R. Pastor-Satorras, E. Smith, and R. Sole (2003) Evolving protein interaction networks through gene duplication. *J. Theor. Biol.*, 222:199–210.

[14] R. Patro, E. Sefer, J. Malin, G. Marcais, S. Navlakha, and C. Kingsford (2011) Parsimonious reconstruction of network evolution. *In Proc. of WABI'11*, LNCS 6833, pp 237–249.

[15] J. Pinney, G. Amoutzias, M. Rattray, and D. Robertson (2007) Reconstruction of ancestral protein interaction networks for the bZIP transcription factors. *Proc. Natl. Acad. Sci.*, 104:20449–20453.

[16] R.Sole, E. Smith, R. Pastor-Satorras, and T. Kepler (2002) A model of large-scale proteome evolutions. *Adv. Complex Syst.*, 5:43–54.

[17] M. Stumpf, W. Kelly, T. Thorne, and C. Wiuf (2007) Evolution at the system level: the natural history of protein interaction networks. *Trends Ecol. Evol.*, 22:366–373.

[18] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani (2003) Modeling of protein interaction networks. *ComPlexUs*, 1:38–44.

[19] A. Wagner (2001) The yeast protein interaction network evolves rapidly and contains few redundent duplicate genes. *Mol. Biol. Evol.*, 18:1283–1292.

[20] T. Yamada and P. Bork (2009) Evolution of biomolecular networks–lessons from metabolic and protein interactions. *Nat. Rev. Mol. Cell Biol.*, 10:791–803.

Appendix

Proof of Lemma 1: Assume that Γ consists of k binary trees $T_1, \cdots, T_k$, and θ is a duplicate sequence compatible with Γ. For each graph G in the graph sequence $\{G_0^\theta, \cdots, G_n^\theta\}$, we can associate it with a graph $\Pi(G)$ as follows. The vertex set of $\Pi(G)$ is $\{1, \cdots, k\}$ and two distinct vertices i and j are adjacent if and only if there exist some adjacent nodes g_i and g_j in G such that g_i is a leaf in the tree T_i and g_j is a leaf in T_j.

Let G be a graph in $\{G_1^\theta, \cdots, G_n^\theta\}$. Denote the anchor node and duplicate node corresponding to this graph by u and v, respectively. Since θ is compatible, u and v are the leaves in the same tree in Γ. Note that for any vertex g that is distinct from u and v, then g is adjacent to u or v in G if and only if g is adjacent to u in $\mathcal{R}_v^u(G)$. Therefore, we can conclude that $\Pi(G) = \Pi(\mathcal{R}_v^u(G))$, and hence also $\Pi(G_0^\theta) = \Pi(G_n^\theta)$. On the other hand, from the construction we know that $\Pi(G_0^\theta)$ is isomorphic to G_0^θ.

In consequence, for two compatible duplicate sequences θ_1 and θ_2, since $G_n^{\theta_1} = G_n = G_n^{\theta_2}$, we can conclude that $G_0^{\theta_1}$ and $G_0^{\theta_2}$ are isomorphic, as required. □

Proof of Theorem 1: We shall establish the lemma by induction on the number of cherries in Γ. The base case that Γ contains no cherry is trivial, because this implies $n = 0$.

Now assume that Γ contains m cherries, and that the lemma holds when the number of cherries in the duplication forest is at most $m-1$. Fix a cherry $\{u, v\}$ in Γ and choose a label g that is not used before. Consider the network G^* that is obtained from $\mathcal{R}_v^u(G)$ by relabeling u with g, and the duplication forest Γ^* obtained from Γ by replacing the cherry $\{u, v\}$ with a leaf labeled as g. Note that either node u or v (possible both) must appear in the duplicate sequence of θ_1; we replace them with g and denote the sequence with the first g removed by θ_1^*. Then θ_1^* is a duplicate sequence that is compatible with Γ^*.

Similarly, the sequence θ_2^* obtained from θ_2 in the same way is also compatible with Γ^*. Now the induction assumption implies $\delta(\theta_1^*) = \delta(\theta_2^*)$. Together with

$$\delta(\theta_1) - \delta(\theta_1^*) = \delta(\theta_2) - \delta(\theta_2^*),$$

we have $\delta(\theta_1) = \delta(\theta_2)$, as required.

On the other hand, the number of edges increased from G_{i-1}^θ to G_i^θ is given by $\delta(v_i)$ and $\alpha(v_i)$, where v_i is the duplicate node. Together with Lemma 1, this implies

$$\delta(\theta_1) + \alpha(\theta_2) = |E(G_n)| - |E(G_0^{\theta_1})| = |E(G_n)| - |E(G_0^{\theta_2})| = \delta(\theta_2) + \alpha(\theta_2).$$

Since $\delta(\theta_1) = \delta(\theta_2)$, we have $\alpha(\theta_1) = \alpha(\theta_2)$. □

Proof of Theorem 2: Let $\theta = \{v_1, \cdots, v_n\}$ be a duplicate sequence that is compatible with the duplication forest Γ. By Lemma 1 and Theorem 1, it is sufficient to note that

$$L(\theta \,|\, G, \mathcal{M}, \Gamma) = p_c^{\delta(\theta)} p^{\alpha(\theta)} q^{\beta(\theta)},$$

holds with $q := (1-p)/2$, an observation following from that

$$\mathbb{P}(G_i^\theta \,|G_{i-1}^\theta, \Gamma, \mathcal{M}) = p_c^{\delta(v_i)} p^{\alpha(v_i)} q^{\beta(v_i)}$$

holds for each $i \in \{1, \cdots, n\}$. □

Proof of Lemma 2: Clearly, we have $G_i^{\theta_1} = G_i^{\theta_2}$ for $i > m$. To show this also holds for $i < m$, it suffices to show $G_{m-1}^{\theta_1} = G_{m-1}^{\theta_2}$ For $i \in \{m, m+1\}$, let u_i be the anchor node of v_i. Since θ_1 and θ_2 are both compatible with Γ, we know that $\{u_m, v_m\}$ and $\{u_{m+1}, v_{m+1}\}$ are two distinct cherries in $\Gamma_{m+1}^{\theta_1} = \Gamma_{m+1}^{\theta_2}$. Therefore, we have

$$\mathcal{R}_{v_m}^{u_m}(\mathcal{R}_{v_{m+1}}^{u_{m+1}}(G_{m+1})) = \mathcal{R}_{v_{m+1}}^{u_{m+1}}(\mathcal{R}_{v_m}^{u_m}(G_{m+1})),$$

because the four nodes u_m, v_m, u_{m+1} and v_{m+1} are distinct. □